

# HYPertext-BASED DESIGN OF A USER INTERFACE FOR SCHEDULING

Irene Wong Woerner  
Eric Biefeld

Jet Propulsion Laboratory  
California Institute of Technology  
Pasadena, CA 91109  
irene\_woerner@isd.jpl.nasa.gov  
biefeld@ai.jpl.nasa.gov

## Abstract

Operations Mission Planner (OMP) is an on-going research project at the Jet Propulsion Laboratory that utilizes Artificial Intelligence techniques to create an intelligent, automated planning and scheduling system. The information space reflects the complexity and diversity of tasks necessary in most real-world scheduling problems. Thus the problem of the user interface is to 1) present as much information as possible at a given moment and 2) allow the user to quickly navigate through the various types of displays. This paper describes a design which applies the hypertext model to solve these user interface problems. The general paradigm is to provide maps and search queries to allow the user to quickly find an interesting conflict or problem. Then allow the user to navigate through the displays in a hypertext fashion.

## 1. Introduction

Planning and scheduling systems address complex tasks and relationships. Most users of scheduling systems are expert schedulers. These users are knowledgeable in their task domain, but are sometimes novices in the underlying computer concepts. Designing usable and effective user interfaces for such systems is very difficult, because while the domain is inherently complex, the user interface should be easy to master and demand easy cognition. Through frequent use, most expert users evolve to be "power" users who are familiar with syntactic and semantic aspects of the system. These users require rapid response times, brief and less distracting feedback, and capability to carry

out actions with a few keystrokes or selection. <sup>1</sup> Design of user interfaces (UI) for scheduling and planning tools need to support transition from novice to power user effectively.

Key issues for a usable UI include presentation of different perspectives for a large and complex information space, presentation of relationships between data units, presentation of an ever changing information space while limiting disorientation for users, navigation through this information space and methods for directed search. Many of these issues lend themselves well to a UI design that is hypertext-based. <sup>2</sup>

## 2. Background

### User Interface Design

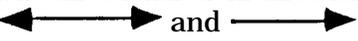
There are many guidelines from many organizations for many types of systems. While some of these guidelines are platform and implementation dependent, this paper will only address general guidelines for data display of an information system UI. Guidelines for display of data are as follows. <sup>3</sup>

- Consistency of data display - Provide consistency in use of terms, abbreviations and formats.
- Efficient information assimilation by a user - Utilize a format familiar to a user and related to tasks required to be performed within this data (i.e., Gantt charts). Also includes use of neat columns for data, left justification for alphanumeric data, right justification for integers, lining up decimal points,

proper spacing, comprehensible labels, and appropriate use of coded or color values.

- Minimal memory load on a user - This includes no requirement for users to remember information from one screen for use on another, arrangement of tasks for completion within a few commands, minimizing opportunities to forget a step, use of common formats and providing labels.
- Compatibility of data display with data entry - Establishing a clear link between the format for data entry and format of displayed information.
- Flexibility for user control of data display - Provide access to information in the form most convenient for the task a user is working on,

## Hypertext

The term "hypertext" is attributed to Ted Nelson, one of the pioneers of hypertext research. As a concept, hypertext has been around for several decades, but useful implementations of hypertext was not realized until greater computing capabilities and more advanced techniques of computer science were widely available. Within the past ten years, hypertext technology has matured considerably, and many commercial systems and tools exist. Hypertext is a technology that allows non-sequential access to data. The hypertext model contains interlinked pieces of information. Each piece of information is called a "node." Nodes have pointers to other nodes, and these pointers are called "links." Links can be mono-directional or hi-directional as depicted by  and  respectively. Anchor nodes depict points of origin and destination nodes depict points of arrival. Relationships between anchor and destination nodes can be one-to-one, one-to-many, many-to-many, and many-to-one.<sup>4</sup> Hypertext systems are typically extensible and editable. Users can add new links or nodes, edit existing links or nodes, and delete links or nodes. Figure 1 is a graphical depiction of the hypertext model. Hypermedia is multimedia hypertext, Multimedia is a term describing diverse information types.

Typical types of media in multimedia are text, graphics, video and sound.

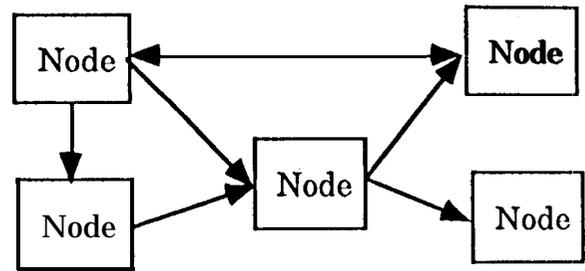


Figure 1: Hypertext Model

Since the primary purpose of hypertext is to permit fast and easy access to large amounts of information by establishing links between related pieces of information, managing complexity and presentation of this model is a major issue for design of any hypertext system.

Studies [Nielsen and Lyngbaek1990] have shown that there is a risk of users becoming disoriented and experiencing difficulty in finding the information they need while moving around a large information space. This phenomenon is referred to as "lost in hyperspace." Techniques to remedy this disorientation include use of overview diagrams or graphical browsers for navigation, at a glance identification of node class, limiting dataspace transitions for less disorientation, retrace mechanisms, well-marked paths for browsing, "link-blazers" and "link followers" (i.e., preview and retrace), and bookmarks for identifying nodes a user may want to return to later. Bookmarks can also include the state of a hypertext system and path information.

## OMP

Operations Mission Planner (OMP) is an ongoing research project<sup>5</sup> at the Jet Propulsion Laboratory. The goal of OMP is to create an intelligent scheduling tool using Artificial Intelligence, Operational Research, and User Interface techniques. Currently, OMP is being used to schedule the 26 Meter Subnetwork of the Deep Space Network (DSN). The examples in this paper are taken from the 26 Meter Subnetwork domain.

OMP decomposes a schedule into two primary object types: activities and resources. An activity represents the requests for services while the resources represent the facility being schedule. In the DSN domain a tracking pass is an activity while an antenna is a resource,

An activity is a temporal interval which reserves a set of resources. Graphically OMP displays activities as a Gantt chart. An activity can include a set of time windows in which it can be scheduled, temporal relations to other activities, priority, and other attributes. For example, a tracking pass has to occur while it can be seen by the antenna (view period) and is followed by a playback,

A resource is a timeline which tracks the time varying availability of some facility. A resource timeline also tracks which activities are using the resource at any point in time. A resource timeline is composed of a temporal series of resource segments. Each resource segment is a temporal interval which records the availability and activities using the resource during the segment temporal interval. As activities are scheduled the number and temporal duration's of the resource segments change.

Graphically OMP displays a resource as a histogram of the resource usage. However, instead of using height to show the resource usage, OMP's resource strip charts use color coding. For example, each antenna in the DSN domain is a resource where background gray represents no usage, green represents in use, and red represents conflict. When a tracking pass is scheduled, the appropriate antenna timeline is updated to show that the antenna is being used during the tracking pass temporal duration. If more than one tracking pass is scheduled on the same antenna at the same time then the antenna resource "shows the conflict for the temporal overlap of the tracking passes.

OMP supports both user editing of a schedule and automatic schedule revision.<sup>6</sup> The user can either invoke a scheduling strategy which will update the existing schedule or use OMP

as a constraint checking schedule editor. The scheduling strategies can either construct a schedule or minimize the conflicts in an existing schedule. Since OMP uses a variety of scheduling strategies interleaved with user editing the interface must both support the user editing of the schedule and allow the user to monitor the strategies.

Tracking resource conflict is a central part in constructing and maintaining a schedule. A scheduled is constructed work by first roughing out a schedule and then by resolving any resource conflicts. After a scheduled is constructed but before it is executed new request or changes in the availability of the resources will necessitate modifications to the schedule. The new requests are added to the existing schedule which usually creates new resource conflicts which must be eliminated.

OMP automatically tracks the relationships between activities and resources. Whenever an activity is scheduled or modified OMP updates the appropriate resource timelines. Both the Gantt charts and the resource strip charts are incrementally updated. The OMP resource timelines can be queried to find any type of resource conflicts or availability. For example, a user clicking on a resource strip chart causes OMP to identify the activities using that resource at the selected time. The animation of the activity Gantt charts and the resource strip charts and the ability to query these charts form the basis of OMPS GUI which allow the user to quickly locate conflicts,

The Gantt charts and the strip charts are displayed in the Timeline Display Window as shown in Figure 2. The various charts in this window show time in the horizontal axis while the charts are stacked vertically. The window can be temporally zoomed in and temporally panned. The temporal zooming and panning effects only the temporal part of the display not the charts labels. The temporal zooming is quantized to logical temporal units, such as a week to two days to 12 hours.



The second major window is the Activity Information Window shown in Figure 3. This displays selected activities in a speed sheet layout. Each row is a separate activity. The columns are the attributes of an activity including meta attributes such as deleted or locked. This display allows the user to directly edit an activity. By using the delete column the user can either delete or undelete an activity. The lock column protects the activity from edits, batch modifications, or strategies modifying the activity.

Depending on the application domain the number of resources runs between a dozen and sixty while the number of activities runs between one hundred and a thousand. It is usually impossible to display all the activities and all the resources at the same time. A major point of this paper is how to deal with this problem.

Usually a Gantt chart shows all the activities within a given project. There are usually ten to sixty projects within a domain. Another common type of Gantt chart displays all the activities using a particular resource. A Gantt chart can also virtually separate activities according to some criteria. For example, a Gantt chart can vertically sort the activities by project for all activities using a particular antenna or they could be sorted by priority. Besides using various selection criteria a Gantt chart or even a resource strip chart can display additional information. The possible observation periods can be plotted on a project Gantt chart. Some of the strategies locate resource bottlenecks which can be plotted on a resource chart. As a result several different types of charts can be designed for a given domain.

The different types of charts can be considered as different views of the same OMP objects. The different views organize the information about the schedule in different ways. Some of the views show lots of detail but needs lots of screen space. Other views show little detail but can give a good overview of the schedule. Which view is appropriate depends on what the user is doing and constantly changes. The user will probably want several different views active at the same time.

The early versions of OMP separated the Gantt charts and the strip charts on the Timeline Display Window and used scroll bars to control the display. The user spent a lot of time searching for the right chart. They also wanted to reorder the charts on the fly. Scrolling through several hundred activities is absurd so a separate activity edit menu was popped up when ever a user selected an activity through a Gantt chart.

When using the system the user spent a lot of time in traversing the various displays in a non-sequential fashion. First all the user would zoom out (or pan over most of the schedule) to find an interesting resource conflict. The user may also have had to scroll through many of resource strip charts to find the conflict. The user would then zoom in and pan the chosen resource conflict. The user would then query the resource chart to identify the involved activities and then scroll to the appropriate Gantt charts. After selecting an activity (or two) to edit the user would then zoom out to find a good place to position the activity. The user then needed to identify the surrounding activities and bring up activity edit menus so the end (or start) times could be copied to the conflicting activity. After the activity has been edited the user would check to see if new conflicts were created, close the open activity menus, and find a new conflict.

Thus the problem of the user interface is to 1) present as much information as possible at a given moment and 2) allow the user to quickly navigate through the various types of displays. The general paradigm is to provide maps and search queries to allow the user to quickly find an interesting conflict or problem. Then allow the user to navigate through the displays in a hypertext fashion. Instead of scrolling through the various charts and panning through time the user could select an item from on display and go to the appropriate view and temporal period.

### 3. Application of Hypertext Model to OMP

An OMP hypertext node is an OMP object from a specific view. Typical nodes are resource segments, or an activity depicted

either as a Gantt or as an activity information row. 1 Hyperlinks are used for navigation to other nodes. Link types are "goto" only. An anchor node is typically an OMP object (i.e., resource segment or an activity). Anchor nodes can have multiple destination nodes. These destination nodes are the cross product of related OMP objects by their possible views. An example of an anchor node and its associated destination nodes is depicted in Figure 4.

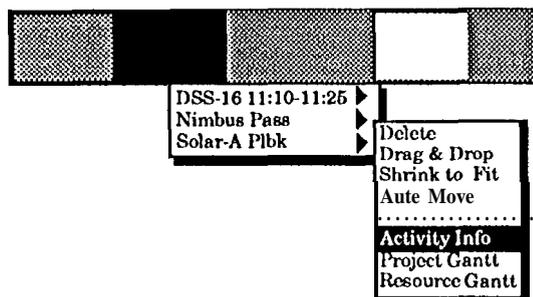


Figure 4: Selecting Destination Node

The design chosen for this is hierarchical menus for different possible destination nodes. The reason hierarchical menus were chosen as opposed to buttons was to optimize use of screen real estate, allow flexibility of cross products between objects and views, and reduce mouse input to speed navigation through nodes. Note that the first or default object, presented in the hierarchical menu is the object itself. This allows users to select different views for this object (i.e., different destination nodes). Also, destination nodes for any anchor node is dependent only on an anchor's object and not its view.

Although OMP maps to the hypertext model, several features of this domain require different node and link representation that differ from traditional hypertext systems. One of the major differences is the dynamic nature of hyperlinks and destination nodes. In traditional systems, the bounds of hyperspace are static or change slowly. Additionally, hypertext authoring is performed at discrete time intervals, and links basically remain static after authoring, and are well defined. Within OMP, the bounds of hyperspace fluctuate constantly. These changes include activities or classes of activities that are continuously added and deleted. Activities

can also use different resources. For example, a tracking pass activity may switch antennas. Additionally, the temporal interval of an activity can change. For example, tracking a satellite during its 38th orbit rather than the 36th. All of these activity fluctuations imply changes in resource segments. Therefore, destination nodes and the bounds of this hyperspace are constantly changing.

One of the rules of clarity in hypertext systems is minimization of transitions between information spaces. Although OMP's information space is continuously changing, we need to minimize disorientation to a user due to these changes. Imagine a Gantt chart which has activities with buttons. These buttons are constantly appearing and disappearing, and can cause disorientation and confusion to a user. In our design, we navigate through these dynamic links by using a pop up menu that is configured on demand. This reduces a user's chance of disorientation in hyperspace. Since OMP's information space is so dynamic, we have made these transitions seamless through the use of pop up menus. By using non-modal windows (i.e., windows that do not persist beyond the current action and require input from a user before any additional action can proceed). In this case, release of the mouse button is the required action. The primary benefit of this method is a snapshot representation of underlying dynamic actions that do not require some animated display. In OMP, therefore, views are animated and nodes are not.

One of the features in OMP is the capability for a user to edit an activity or to invoke a scheduling strategy which will automatically edit several activities. When this strategy is invoked, there are several display options. One option is to blank the views while the strategy is running, and notify and refresh the display when the strategy is complete. This option saves the most CPU cycles. Another option is to animate the display views. This option provides a user with some feedback on the strategy's progress and actions. The final display option is to display the associated hypertext links and nodes (i.e., the system determines path traversal during execution of a scheduling strategy). For example, the

engine may schedule an

activity to resolve a resource conflict. If this resource is not visible as a resource timeline, the system will invoke the view and page to the appropriate time interval. In this option, the system is animating default navigation paths. This allows a user to carefully monitor actions and logic of a scheduling strategy which is mainly used by developers to debug or tune a strategy.

histories, because the depth of paths is typically less than five traversals. This broad nature of hyperspace requires a navigation map that can encapsulate an overview or summary for the entire information space. Since the timeline is an integral representation for scheduling, a temporal overview is ideal for navigation assistance. Two possible temporal map designs are

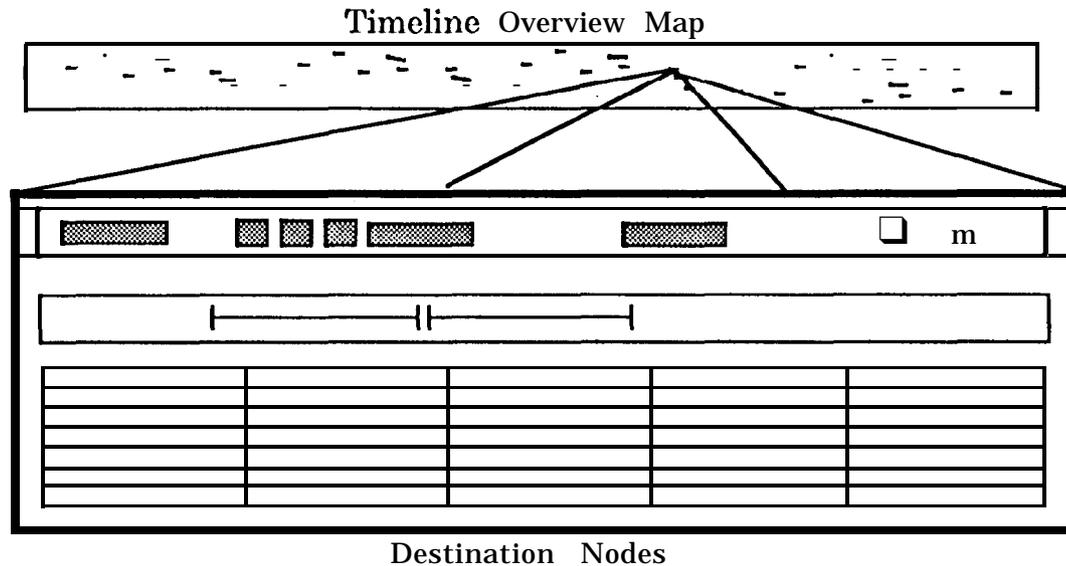


Figure 5: General Temporal Navigation Map

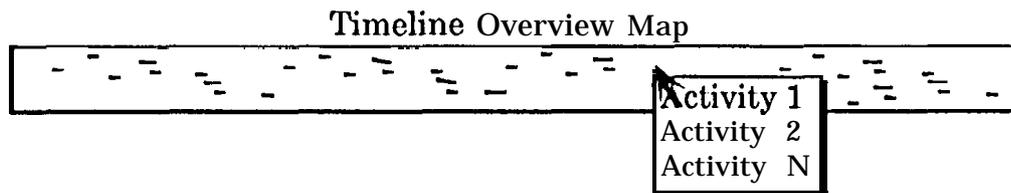


Figure 6: Fisheye Temporal Navigation Map

OMP nodes are composed of resource segments, activities in Gantt form or descriptions of activities, and displays depicting this information are almost always visible. In OMP, paths of linked nodes are not rich in depth, but in breadth. The notion of retrace paths or maintaining path traversal history was examined and abandoned. With multiple windows open, there is not much value added for maintaining exact path

considered. A general overview map as shown in Figure 5. and a fisheye view as described by Furnas in Figure 6.7

With a general design, clicking on the anchor node (i.e., point of interest) displays appropriate destination activities information, Gantt, and resource segment nodes,

Fisheye views provide an overview and more detailed information immediately surrounding the area of interest to show context.

Information that is further away is in summary form. This view can accommodate two potential properties of this information space, 1) estimation of distance (i.e., time) between any given location and the current focus of interest and 2) displaying information at more than one level of detail. In this design, holding the mouse down brings up an annotation window listing activities in that area, and double clicking activates hypertext links displaying activities information, Gantt, and resource segment nodes as in Figure 5.

Bookmarks in typical hypertext systems allow users easier access to destination nodes they may want to return to in the future. For example, after a user finds an activity to move, the user searches for a good location for the activity. The user then goes back to the activity, selects the activity, and then goes back to the good location.

As stated previously, the OMP domain requires short path traversals and the space is continuously changing. Since the space is continuously changing a book mark is not as well defined as in conventional hypertext systems. In OMP a bookmark is tied to a chart and a temporal moment. The user can optionally assign a label to a bookmark. If the user does not assign a label, then the bookmark is stored in a menu of the most recent seven unlabeled bookmarks.

A more interesting application for bookmarks is as short cuts for directed searches. Queries are a particular view of the information space. For example, a search for all conflicts with a particular resource. Since the actual combinations of links and nodes are constantly changing, the concept of the well-marked path to a particular destination is not the actual links and nodes, but the "route" of this path for conceptually related pieces of information. Figure 7 illustrates how queries are performed in OMP.

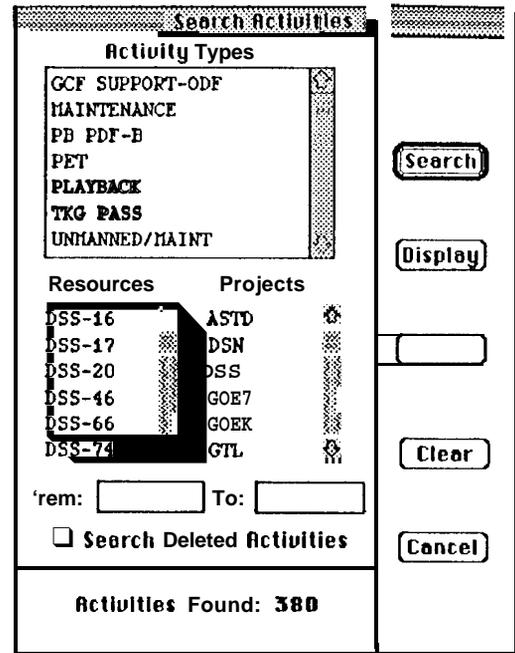


Figure 7: OMP Query Window

The "Save As" button in this window allows users to save query requests as bookmarks to return to the same view in this information space at some later time. Results of queries are returned and displayed in the same summary format of temporal navigation maps as shown in Figure 8. This map is labeled with a user assigned or system assigned name, and functions identically as general navigation maps. This query results map provides a filtered view of possible destination nodes in hyperspace.

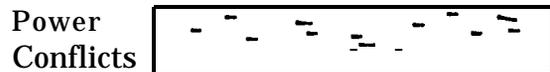


Figure 8: Query Result Navigation Map

## 5. Conclusions

Many OMP UI issues, such as mix mode scheduling interactions, still need to be explored. However, a hypertext-based design as described in this paper will help users in accessing information from a dynamic and complex space more effectively.

## Acknowledgements

The authors wish to thank Barbara Holder of the Advanced User Interfaces Group for her concept of applying hypertext to scheduling interfaces. This work was performed at the Jet Propulsion Laboratory, California Institute of Technology under contract to the National Aeronautics and Space Administration.

## References

- 1 Shneiderman, B. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Addison-Wesley Publishing Company, Inc., 1987.
- 2 Gygi, K. "Recognizing the Symptoms of Hypertext." *The Art of Human-Computer Interface Design*. Addison-Wesley Publishing Company, Inc., 1990.
- 3 Smith, S. and Mosier, J. *Design Guidelines for the User Interface for Computer-based Information Systems*. The MITRE Corporation, 1984.
- 4 Nielsen, Jakob. *Hypertext and Hypermedia*. Academic Press, Inc., 1990.
- 5 Biefeld, E. and L. Cooper "Scheduling with Chronology-Directed Search" *Proceedings of AIAA Computers in Aerospace VII*, 1989.
- 6 Cooper, L. and E. Biefeld "User Interface Issues in Supporting Human-Computer Integrated Scheduling" *Proceedings of the Space Operations, Applications, and Research Symposium (SOAR)*, 1990.
- 7 Furnas, G.W. "Generalized Fisheye Views," *Proceedings ACM CHI'86*. Boston, MA, 1986.